



UTN – FRC

MECATRONICA

PASANTIAS Y SEMINARIOS

SEMINARIO DE PRESENTACION JAVA

Torres, Horacio Gonzalo
108832


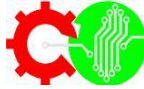
 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba					
2023		Tecnicatura Universitaria en Mecatrónica			
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios		
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

INTRODUCCION

El presente informe plasma la temática desarrollada y profundizada en el Seminario de Introducción a Java dictado el 24 de junio de 2023 como parte de la materia Pasantías y Seminarios de la presente Tecnicatura.



Se espera que los asistentes de este seminario puedan conocer la tecnología de la JVM, sus ventajas y funcionalidades principales.

Este seminario no es un curso de formación, sino una presentación, con intención de brindar un panorama general del lenguaje, destinado a una audiencia con conocimientos básicos de lógica de programación.

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba		 Tecnicatura Universitaria en Mecatrónica			
2023		Pasantías y Seminarios			
Docentes	Carlos Centeno – Karina Corvalan	Materia:			
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

INDICE

Introducción.....	1
Índice.....	2
Anteproyecto	
Presentación.....	3
Temario.....	3
Preparación de la Presentación	
Fuentes.....	4
Elección de Temas.....	4
Objetivos.....	5
Desarrollo de la Presentación.....	6
Bitácora.....	8
Conclusión.....	9
Referencias.....	10
Apéndice – Presentación PPT.....	11

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba					
2023		Tecnicatura Universitaria en Mecatrónica			
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios		
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

ANTEPROYECTO

Presentación:

Siendo el tercer lenguaje más utilizado en este momento, y con una vigencia de más de 25 años, Java está presente en multitud de dispositivos y aplicaciones, siendo usado a diario por la mayoría de las personas, muchas veces sin siquiera notarlo.

La independencia del Sistema permite reutilizar código fácilmente, reduciendo tiempos y costos de desarrollo. Los modificadores de acceso y el lenguaje fuertemente tipado permiten proteger y gestionar información confidencial. El sistema de archivado JAR permite crear y compartir librerías y aplicaciones. Se pueden utilizar gestores de dependencias para facilitar el mantenimiento de una aplicación y empaquetar librerías.



El conjunto de estas funcionalidades, sumado al acceso libre al entorno de ejecución (JRE) y al entorno de desarrollo (JDK) convierten a Java en una herramienta potente y versátil para el desarrollo de aplicaciones, APIs, y el uso combinado con otros lenguajes para mejorar la experiencia de usuario.

Temario:

Con el objetivo de mostrar el potencial de la plataforma y generar un interés en posibles desarrolladores, se presentarán la siguiente lista de temas a tratar:

- Historia y evolución del lenguaje - B, C, POO
- Que es java? Funcionalidad y ventajas de la máquina virtual
- Jre, jdk, comentarios, javadoc
- Seguridad, ventajas, desventajas
- Aplicaciones – consola – app – api – webapps (web2)
- Aprendizaje - rutas

Con una extensión de aproximadamente 15 a 20 minutos por bloque, se estima una presentación con una duración total aproximada de 1 hora y media a 2 horas, con un bloque final para consulta.

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba					
2023		Tecnatura Universitaria en Mecatrónica			
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios		
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

PREPARACION DE LA PRESENTACION

Fuentes:

Para la presentación se extrajo material de diversas fuentes, siempre teniendo la condición de que las fuentes sean open source para mantener el contenido libre y replicable. También se importaron imágenes de presentaciones de clases del curso “Introducción a Java” de Argentina Programa 4.0 y del curso “C Programming – A Historical Perspective” y “Python for Everybody” del Dr. Charles Severance. Para el resto de las imágenes se utilizaron fuentes de acceso público como Google y Wikipedia. Cabe mencionar también que se utilizó IA de gamma.app para generar una presentación genérica, la cual sirvió de base para diagramar y extender la presentación final, y prompts de ChatGPT para llenar las lagunas de contenido donde fue necesario. Si bien la presentación generada de esta forma no es suficiente por si misma, el potencial de la IA es enorme; en palabras de ChatGPT:

“Si bien la inteligencia artificial aún no ha alcanzado la capacidad total para realizar presentaciones de manera autónoma, es innegable que posee un potencial significativo. Aunque los modelos de lenguaje como yo, ChatGPT, han sido entrenados para generar respuestas coherentes y útiles, todavía existe un margen de mejora en áreas como la fluidez del discurso y la adaptabilidad contextual.”

Las fuentes utilizadas se mencionan más adelante con sus respectivas referencias para consulta.


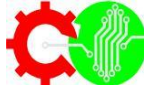
Elección de Temas:

Si bien la presentación es sobre el lenguaje Java, los temas tratados se aplican a cualquier lenguaje fuertemente tipado y con orientación a objetos. Dicho de otra forma, la presentación intenta mostrar los procedimientos, ventajas y lógica que nos ofrece el Paradigma Orientado a Objetos (POO), usando el lenguaje Java para plasmar esos conceptos en código real; una vez se tengan los conceptos entendidos y afianzados, se pueden aplicar con relativa facilidad en cualquier lenguaje orientado a objetos, ya sea Java, C#, Python, JavaScript, etc.

El concepto de maquina virtual o interprete es ampliamente utilizado en los lenguajes orientados a objetos. Este interprete permite la portabilidad del código, haciéndolo independiente de la plataforma donde se esté ejecutando.


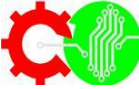
POO define también términos como encapsulamiento, herencia, polimorfismo y clases abstractas, los cuales son muy útiles a la hora de desarrollar una aplicación.

Finalmente, se intenta poner regularmente en perspectiva el hecho de que todos los lenguajes superiores son interpretados a C, por lo que todos los conceptos y funcionalidades presentadas se podrían escribir directamente en código C a través del uso de librerías; si bien este método no es practico, considero que vale tener presente que todo lo que programamos termina siendo convertido en señales lógicas almacenadas en transistores microscópicos, que son traducidas una determinada cantidad de veces de forma de poder ser interpretadas tanto por la maquina como por la persona.

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba		 Tecnicatura Universitaria en Mecatrónica			
2023		Pasantías y Seminarios			
Docentes	Carlos Centeno – Karina Corvalan	Materia:			
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

OBJETIVOS

- Introducir a los oyentes en el lenguaje Java, junto con sus funcionalidades principales y los conceptos básicos de POO.
- Mostrar ejemplos prácticos de uso, a través de aplicaciones y live coding.
- Generar interés y alentar al desarrollo de aplicaciones y la programación en general.

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba			
2023		Tecnicatura Universitaria en Mecatrónica	
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios
Alumno:	Horacio Gonzalo Torres	Legajo:	108832
		Curso:	2M2

DESARROLLO DE LA PRESENTACION

La presentación comienza con una breve historia de los lenguajes mas utilizados. El primer lenguaje fue conocido como B, el cual tenía la característica de ser lo suficientemente compacto como para ser almacenado en las computadoras de los '60 sin afectar notablemente su capacidad. Luego se avanza a C, el cual permite varios tipos de datos, y cuyo núcleo de procesos es casi independiente del equipo donde este corriendo. Estas dos características lo convierten en un lenguaje eficiente y portable, ideal para ser usado como sistema operativo, por lo que aun hoy está vigente en prácticamente la totalidad de los equipos en funcionamiento. A menudo se hace referencia a C como un lenguaje "close to the metal" debido a su manejo de recursos a bajo nivel, su portabilidad y eficiencia. Se hace foco en que (casi) todos los programas en uso actualmente están programados en C, o están programados en un lenguaje que fue programado en C. Se avanza hacia el POO, el cual tomó relevancia en los '90, cuando la experiencia de uso cambió de la lectura por consola a una interfaz grafica (Windows). Se mencionan y explican brevemente los pilares sobre los que fue construido (abstracción, herencia, polimorfismo y encapsulamiento). Finalmente, se presenta una línea temporal con los principales lenguajes usados, desde Assembler pasando por C, hasta Java y C#.

Luego de esto, se presenta Java como lenguaje propiamente dicho. Se toma en relevancia el hecho de que es un lenguaje compilado a un código común, y luego interpretado a C a través de una maquina virtual. Se mencionan ventajas del lenguaje y los pilares que inspiraron la creación del lenguaje. Se menciona el concepto WORA (Write Once Read Anywhere), que es la base de la portabilidad de código. Java es un lenguaje robusto, seguro y orientado a objetos.



Se indica la diferencia entre JRE (Java Runtime Environment) y JDK (Java Development Kit). JRE contiene la maquina virtual y todo lo necesario para correr programas en Java, mientras que JDK contiene el JRE y el compilador, necesarios para generar el código que va a ser interpretado por la maquina virtual. Se hace énfasis en que pueden cohabitar distintas versiones de Java, instalando distintos JRE y JDK; se indica también las versiones actuales de JRE y JDK.

Luego se explica el concepto de maquina virtual; la JVM es una computadora virtual, que utiliza la memoria del equipo donde se este ejecutando para cargar y procesar datos basados en el programa que se esté corriendo. El código se escribe en formato Java (.java) y se compila a través de javac al formato .class; este archivo es leído por la maquina virtual e interpretado en el sistema operativo de la maquina real. Se hace énfasis en que el archivo .class es igual para todos los sistemas operativos, siendo la maquina virtual la responsable de "traducir" este código al archivo binario que va a ejecutar la maquina real, el cual depende del sistema operativo y (por extensión) del hardware instalado en la máquina. Se explica el funcionamiento del stack de memoria, donde se guardan las referencias a los objetos y de la memoria heap, donde se guardan los objetos. Se explica el uso del garbage collector (recolector de basura) y el concepto de memory leaks (pérdidas de memoria).

Se indican brevemente los tipos de datos. Estos se dividen en Primitivos, que son los heredados de C, y los tipos Objeto, indicando ejemplos de Objetos (clases) ya incluidos en el lenguaje, Arrays y Wrappers (envoltorios), que son la versión Objeto de las clases Primitivas. Se indican las diferencias en el uso de memoria de datos Primitivos y de Objetos.

Se desglosa la estructura dentro de un proyecto Java. El código fuente se almacena dentro de la carpeta Source, la cual puede contener diferentes paquetes, que son carpetas donde se almacenan clases. En las clases se escribe el código, ya sea la estructura de objetos, o los métodos o funciones que utilizara el proyecto. Se define una de las clases como clase main, que es la clase principal desde donde son llamadas todas las clases y métodos del proyecto. Se explica la composición de una clase, desde el paquete donde esta incluida, pasando por los atributos o propiedades, el constructor de clase y los métodos que incluye, explicando getters y setter, modificadores de acceso y la etiqueta @Override, que lleva a explicar los conceptos de herencia y polimorfismo. Se explica el concepto de sobrecarga de método, al escribir más de un método con el mismo nombre, pero modificando los argumentos y la salida del método. Se hace énfasis en que toda clase debe tener un constructor para poder instanciarse (no se puede crear un objeto si este no tiene un constructor en su clase).

Se explica la jerarquía de clases, la cual permite que una clase pueda heredar de su clase padre a través del modificador "extends"; se indica que una clase solo puede tener un padre, y que existe una clase "Object" que es la clase padre de todas las clases. Para los casos donde sea útil implementar métodos de mas de una clase, se introduce el concepto de Interface y clases abstractas; si bien una clase solo puede heredar de un padre, puede implementar métodos de una o varias Interfaces a través del modificador "implements". Se explica cómo diferentes clases pueden tener distintos comportamientos según como se modifique el mismo método a través de un Override, y se hace énfasis

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba			
2023		Tecnicatura Universitaria en Mecatrónica	
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios
Alumno:	Horacio Gonzalo Torres	Legajo:	108832
		Curso:	2M2

en que se deben implementar manualmente todos los métodos contenidos en un Interface, ya que, al no estar atada a una jerarquía de clases, no se heredan, y por lo tanto deben ser implementados explícitamente.

Se mencionan las estructuras de flujo de control en Java, los cuales son iguales a los de C (condicionales y bucles). Se introduce el bloque try / catch para manejo de excepciones. Este bloque se aplica en general cuando hay que leer datos, ya sea desde el teclado o desde una base de datos; al leer el dato puede pasar que éste sea incorrecto o inexistente, entonces el bloque try / catch nos permite que, en caso de falla en el código, se ejecute el bloque catch, permitiendo que el programa se siga ejecutando. Se indica que se pueden tener múltiples bloques catch para distintos tipos de excepciones. Se introduce el bloque finally, el cual se ejecuta independientemente de que se genere o no una excepción.

Se menciona la diferencia entre errores y excepciones, y se introduce la clase Throwable (lanzable). Se citan errores y excepciones comunes, y se explica el concepto con un ejemplo. Se hace énfasis en que es necesario cerrar las conexiones y declaraciones, explicando la funcionalidad del bloque finally.

Se introduce el concepto de Colecciones, explicando las colecciones más comunes (ArrayList y HashMap). Las colecciones son arrays de tamaño dinámico, lo que nos permite desentendernos del tamaño del array al momento de declarar la variable. Se explica la diferencia entre una List, donde los elementos están ordenados por un número entero creciente, y un HashMap, donde los datos se ordenan con un par llave-valor, teniendo la llave la característica de no estar restringida a un valor entero, sino que puede ser cualquier tipo de dato.

Se explica la persistencia de la información, que nos permite almacenar datos generados por el programa y acceder a la información una vez creada. La información se puede guardar en bases de datos o en archivos, nombrándose los formatos más comunes.

Las bases de datos pueden ser relacionales (similares a una hoja de Excel) con tuplas de datos en filas ordenadas por un id (que indica la relación entre distintos datos); o pueden ser no relacionales, las cuales se ordenan en un formato (XML o JSON), y se acceden a través de una combinación de ArrayLists y HashMaps.

Otra forma de almacenar datos es a través de archivos; estos pueden ser de texto plano, PDF, CSV (valores separados por comas), imágenes y archivos binarios en general. La principal diferencia entre archivos y bases de datos es que las bases ya están ordenadas y tipificadas, por lo que es más fácil acceder desde diferentes aplicaciones y dispositivos.



Se explica la utilidad de generar los proyectos a través de un gestor de dependencias, utilizando Maven como ejemplo funcional. Un gestor de dependencias reduce significativamente el tiempo necesario para entender y mantener un proyecto existente (curva de aprendizaje). A través del archivo pom del proyecto, se puede ver la versión de Java en que fue compilado, qué dependencias (librerías) tiene, la versión actual del proyecto y la clase principal, entre otros datos. Se ilustra el concepto con un ejemplo.

Luego se introduce la API JDBC, la cual nos permite conectar una aplicación con una base de datos, utilizando el concepto de intérprete, ya que debemos instalar el driver necesario dependiendo de la base de datos que vamos a conectar, encargándose la API de interpretar las instrucciones del programa y conectar a la base. Se explican las 4 operaciones básicas con base de datos, el acrónimo CRUD (crear, leer, actualizar y borrar), y se indican los pasos necesarios que ejecuta la API para hacer una operación contra la base de datos (registrarse, crear conexión, crear la sentencia, ejecutar la sentencia y cerrar la conexión). Se enfatiza la necesidad de cerrar la conexión y la sentencia al terminar la operación, y se explica brevemente el concepto de inyección de código. Se ilustra el funcionamiento de la API con un diagrama, y se enfatiza el uso de bloques try/catch para cada query.

Se indican brevemente los casos de uso de aplicaciones Java, las cuales pueden ser aplicaciones de consola, aplicaciones de escritorio, o aplicaciones web. Se da énfasis a las aplicaciones API web, las cuales son actualmente las aplicaciones más utilizadas.

Para cerrar la presentación, se indican una serie de rutas para aprender a programar en Java, y se indica el repositorio desde donde se puede descargar la presentación.


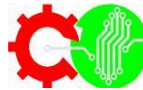
Finalmente se explica mediante un pequeño programa el funcionamiento del compilador javac, y se desglosa una aplicación Winform conectada a una base de datos mediante JDBC, explicando el funcionamiento y los conceptos desarrollados aplicados en el programa.

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba				
2023		Tecnicatura Universitaria en Mecatrónica		
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios	
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso: 2M2

BITACORA

A continuación, se incluye el detalle de horas invertidas en la pasantía, se debe tener en consideración que hubo un cambio de pasantía, iniciando originalmente con un proyecto OTA en C, cambiando luego al proyecto actual.

mes	día	entrada	salida	sumatoria	detalle
mayo				58,0	acumulado con soportes
abril	1	9,0	11,0	60,0	reunión virtual utn - diagramación de pasantías
abril	6	9,0	13,0	64,0	ordenar trabajo previo y diagramar código existente en bloques para análisis
abril	10	15,0	18,0	67,0	revisar librería CRC32, investigar base de checksum y algebra booleana
abril	11	15,0	18,0	70,0	revisar librería FS, investigar clases arduino y tipos y operadores c
abril	12	15,0	18,0	73,0	revisar librerías SPIFFS, comprobar alcance para actualización mediante sd
abril	13	15,0	18,0	76,0	continua revisión spiiffs y yaffs. Comienzo revisión tinyGSM
abril	15	9,0	11,5	78,5	reunión utn pasantías - laboratorio central - cambio de pasantía
abril	17	9	13	82,5	fin revisión tinyGSM. Comienzo revision Update.h
abril	21	9	12	85,5	fin revisión librerías, revisar script - falta completar server
abril	22	8,5	9,5	86,5	reunión virtual utn - progreso pasantías
mayo	6	9	10,5	88,0	reunión virtual utn - progreso pasantías
mayo	17	8	19	99,0	Congreso Latinoamericano en Energías
mayo	18	18	21	102,0	soldado de placas para seminario IoT
junio	2	15	18	105,0	Congreso de Robótica Educativa
junio	6	10	12	107,0	diagramado charla presentación Java
junio	8	10	14	111,0	armado de presentación
junio	10	9	14	116,0	armado de presentación
junio	13	10	15	121,0	armado de presentación
junio	14	15	22	128,0	armado de presentación
junio	15	14	22	136,0	armado de presentación - envío para revisión
junio	24	9,5	12,5	139,0	dictado de seminario
julio	3	9	18	148,0	armado de informe final - envío para corrección

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba					
2023		Tecnicatura Universitaria en Mecatrónica			
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios		
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2


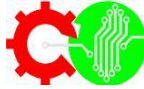
CONCLUSION

La presentación tuvo una duración total de aproximadamente dos horas y media, excediendo en una hora el tiempo estimado previamente.

En cuanto a los objetivos propuestos, se pudieron introducir casi todos los conceptos del temario, quedando fuera solamente los comentarios y javadoc. Si bien estos temas son útiles a la hora de programar, incluirlos hubiera significado extenderse aun mas en el tiempo de presentación, sin aportar al objetivo de introducción al lenguaje.



En cuanto a ejemplos prácticos, se pudo completar un pequeño programa para ilustrar el uso del compilador fuera del IDE (entorno de desarrollo integrado), y se explico de forma más detallada una aplicación conectada a una base de datos, donde se integran todos los conceptos desarrollados en la charla.

Finalmente, no se generó la reacción esperada en la audiencia, ya que, en el transcurso de las aproximadamente 2 horas y media de presentación, solo hubo una pregunta; esto indica que el seminario fue demasiado extenso, lo que genera dispersión en la audiencia y una consiguiente retención baja de conceptos. Como oportunidades de mejora, se puede modificar la presentación para reducir la cantidad de temas expuestos, limitándose solamente a lo necesario para mostrar el lenguaje. También sería muy provechoso un break de 10 a 15 minutos a mitad de la presentación para mejorar la concentración y que la charla no se vuelva pesada.

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba			
2023		Tecnicatura Universitaria en Mecatrónica	
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios
Alumno:	Horacio Gonzalo Torres	Legajo:	108832
		Curso:	2M2

REFERENCIAS

- *Argentina programa – Programador Java Inicial – UTN*
<https://www.argentina.gob.ar/economia/conocimiento/argentina-programa/programacion>
- *Alura ONE – Backend*
<https://app.aluracursos.com/form-one/registro/latam-general>
- *Charles Severance – Python for Everybody – C Programming, A Historical Perspective*
<https://www.py4e.com/>
<https://www.cc4e.com/>
- *Gamma app*
<https://gamma.app/>
- *Chat GPT*
<https://openai.com/blog/chatgpt>
- *Happy Coding*
<https://happycoding.io>
- *Repositorio de la Presentacion – GITHub*
<https://github.com/zalotores/SeminarioJava.git>

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba					
2023		Tecnicatura Universitaria en Mecatrónica			
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios		
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

APENDICE – PRESENTACION PPT

¡Bienvenidos a la presentación de Java!

Java es un lenguaje de programación popular y versátil utilizado en aplicaciones web, móviles, de escritorio y para el Internet de las cosas. Acompañenme en este viaje para aprender más sobre Java.


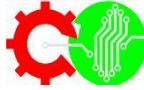



EVOLUCION DE LENGUAJES DE PROGRAMACION

B

C

POO

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba		 Tecnicatura Universitaria en Mecatrónica		
2023		Pasantías y Seminarios		
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios	
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso: 2M2


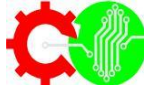


- ◆ Es una simplificación del lenguaje BCPL (Lenguaje de Programación Básico Combinado) capaz de almacenarse en la memoria de las computadoras de la época.
- ◆ Implementado por primera vez en 1969.
- ◆ Tenía un solo tipo de dato (palabra).
- ◆ Operaciones: suma, resta, multiplicación y división. Referencia a punteros.
- ◆ Se debía adaptar a cada arquitectura de hardware
- ◆ En 1973 se implementó el paquete de Entrada / Salida portable, que se convertiría en standard I/O (o stdio) en C.
- ◆ Se utilizó hasta los 90 en algunas máquinas Honeywell y en algunos sistemas embebidos.



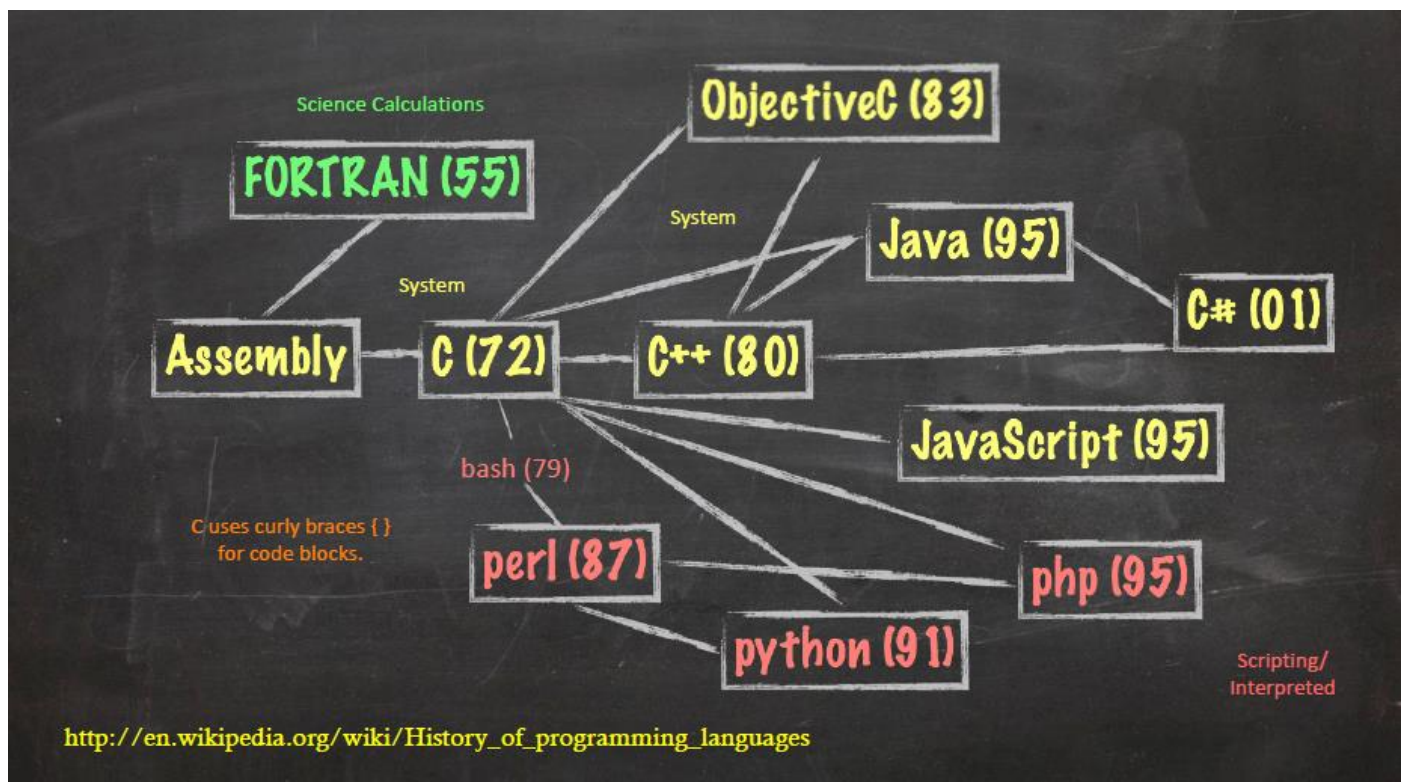


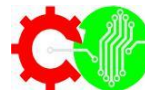
- ◆ Desarrollado entre 1969 y 1972
- ◆ Tipos de datos primitivos int, char, float (basado en byte)
- ◆ Facilidad de implementación y actualización de Hardware mediante core o kernel
- ◆ Se usa en la actualidad para SO y sistemas embebidos en general
- ◆ Se desarrolló junto con Unix
- ◆ C++ desarrollado como forma más potente y flexible (incluye clases)

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba		 Tecnicatura Universitaria en Mecatrónica		
2023		Pasantías y Seminarios		
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios	
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso: 2M2

P00

- ◊ Todo es un objeto
- ◊ Tipos primitivos (wrappers) y tipos objeto
- ◊ Adaptado a interface grafica de usuario
- ◊ Originado en la década del 60, tomó relevancia en los '90
- ◊ Basado en C
- ◊ Pilares
 - ◊ Abstracción
 - ◊ Encapsulamiento
 - ◊ Herencia
 - ◊ Polimorfismo





Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios		
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

Java

- ❖ Creada por Sun Microsystems en 1995 y adquirida por Oracle en 2010
- ❖ Es un lenguaje compilado e interpretado a través de una máquina virtual
- ❖ Gratuito para personas y desarrolladores
- ❖ Lenguaje maduro con una gran comunidad activa
- ❖ Objetivos del lenguaje
 - ❖ Orientado a objetos
 - ❖ WORA (Write Once Read Everywhere)
 - ❖ Robusto
 - ❖ Mejorar la experiencia de uso
 - ❖ Seguridad



Componentes de la máquina virtual

JRE

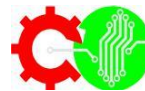
- ❖ Java Runtime Environment
- ❖ Incluye los componentes necesarios para correr aplicaciones en el SO
- ❖ Actualmente en versión 8

JDK

- ❖ Java Development Kit
- ❖ Incluye el JRE
- ❖ Incluye todo el entorno para desarrollar y ejecutar las aplicaciones
- ❖ Actualmente en versión 20



2023



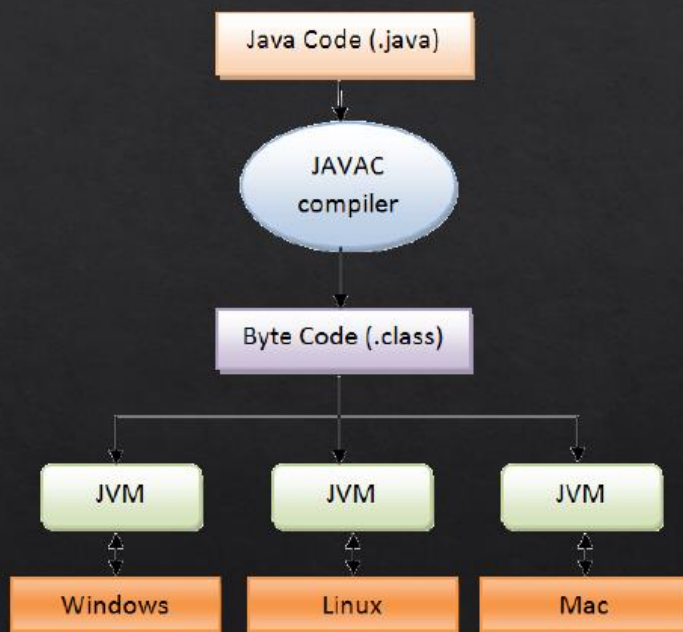
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios		
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

JVM

Se genera el código en .java

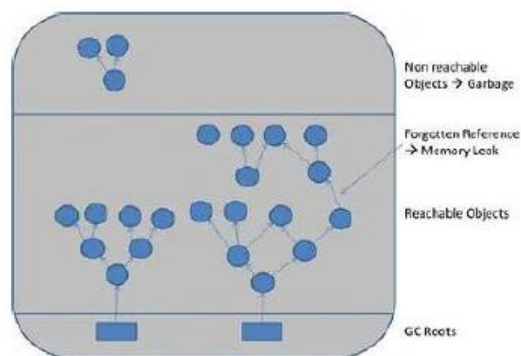
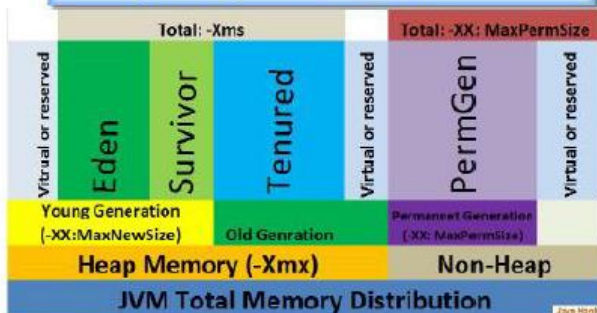
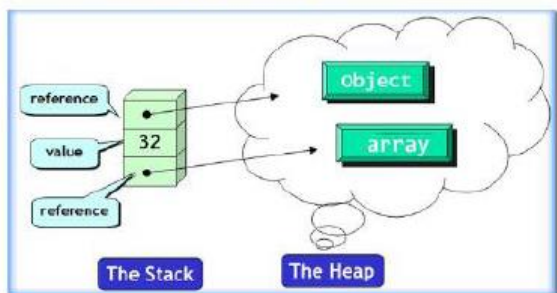
Se compila en un .class


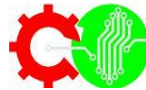
Se ejecuta en .bin mediante
JVM



Java - Memoria y Garbage Collector

Argentina
programa
4.0



 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba					
2023		Tecnicatura Universitaria en Mecatrónica			
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios		
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

TIPOS DE DATOS

Primitivos

- ◆ int
- ◆ long
- ◆ float
- ◆ double
- ◆ char
- ◆ boolean
- ◆ byte
- ◆ short

Objetos

- ◆ Tipos de la Biblioteca Estándar
 - ◆ String
 - ◆ Scanner
 - ◆ System
- ◆ Arrays
- ◆ Wrappers
 - ◆ Integer
 - ◆ Long
 - ◆ Boolean
 - ◆ Character
 - ◆ Double

VARIABLES

Primitivas

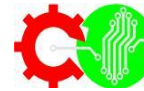
- ◆ Se guarda el valor en la variable
- ◆ Cuando se copia, se copia el valor en una nueva variable

Objetos

- ◆ El nombre de la variable refiere a la dirección de memoria del objeto
- ◆ Cuando se copia, se copia la dirección de memoria del objeto (pointer), por lo que si se modifica la copia se modifica el original



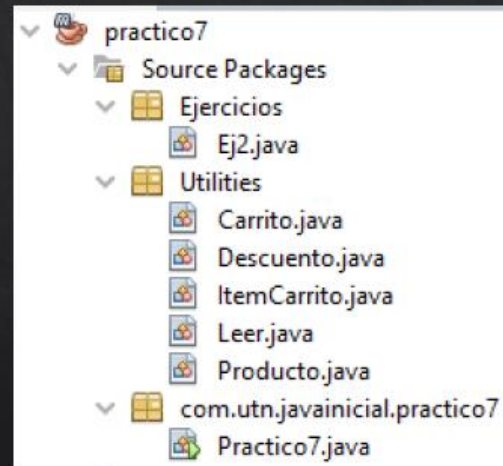
2023



Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios		
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

Estructura de datos

- ◆ Las clases se guardan en paquetes
- ◆ Los paquetes sirven para organizar clases por tipos
- ◆ Los paquetes se guardan en carpeta Source
- ◆ Puede haber otras carpetas (Test, Target)



Composición de Clase

- ◆ Indica a que paquete pertenece
- ◆ Define atributos (encapsulamiento)
- ◆ Define constructor
- ◆ Define métodos (polimorfismo)
- ◆ Sobrescribe métodos heredados (herencia)

```
package Utilities;

/**...4 lines */
public class ItemProducto {

    private int codigo;
    private float precioUnitario;
    private String descripcion;


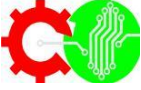
    public ItemProducto(int codigo, float precioUnitario, String descripcion) {
        this.codigo = codigo;
        this.precioUnitario = precioUnitario;
        this.descripcion = descripcion;
    }

    public int getCodigo() {
        return codigo;
    }

    public float getPrecioUnitario() {
        return precioUnitario;
    }

    public String getDescripcion() {
        return descripcion;
    }

    @Override
    public String toString() {
        return "ItemProducto{" + "codigo=" + codigo + ", precioUnitario=" +
            precioUnitario + ", descripcion=" + descripcion + '}';
    }
}
```

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba			
2023		Tecnicatura Universitaria en Mecatrónica	
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios
Alumno:	Horacio Gonzalo Torres	Legajo:	108832
		Curso:	2M2

HERENCIA

- ◆ CLASE PADRE
- ◆ Un objeto solo puede tener un padre (extends)
- ◆ Se puede remontar el árbol de herencia hasta la clase Object
- ◆ Las clases hijas heredan los atributos y métodos de la clase padre
- ◆ Las clases hijas pueden agregar sus propios atributos y clases
- ◆ Las clases hijas pueden sobrescribir (@Override) los métodos de la clase padre (polimorfismo)
- ◆ INTERFACE
- ◆ Es una clase Abstracta (no se puede instanciar)
- ◆ Un objeto puede implementar múltiples interfaces (implements)
- ◆ Un objeto debe tener definido todos los métodos que tenga la interface implementada

Flujo de control en Java

1 Condicionales

Las estructuras if y switch se utilizan para ejecutar diferentes bloques de código según una condición.



2 Bucles

Los bucles for, while y do-while se utilizan para ejecutar el mismo bloque de código varias veces.

3 Control de excepciones

El manejo de excepciones permite que las aplicaciones respondan y se recuperen de eventos inesperados.

<https://gamma.app>

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba		 Tecnicatura Universitaria en Mecatrónica	
2023		Pasantías y Seminarios	
Docentes	Carlos Centeno – Karina Corvalan	Materia:	
Alumno:	Horacio Gonzalo Torres	Legajo:	108832
		Curso:	2M2

Errores / Excepciones

Errores

- ❖ Son detectados por el compilador
- ❖ Normalmente evitan el funcionamiento de una aplicación desde el inicio

Excepciones

- ❖ Puede no ser detectado por el compilador
- ❖ Se generan cuando el programa tiene que procesar un dato inesperado
- ❖ Puede ser atrapado y manejado con un bloque try/catch (RuntimeException)
- ❖ Se pueden generar excepciones personalizadas (Throwable)

Excepciones Comunes

Java y algunas librerías base, ya establecieron excepciones comunes y algunas de ellas deberían ser utilizadas o heredadas antes de crear otras.

No chequeadas

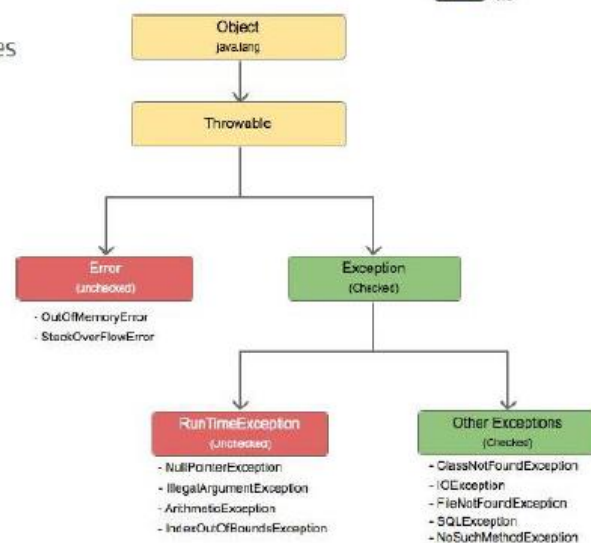
- NullPointerException
- DivisionByZeroException
- IllegalStateException

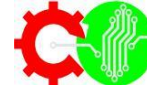
Chequeadas

- FileNotFoundException
- IOException

....

Argentina
programa
4.0





Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios		
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

Manejo de Excepciones

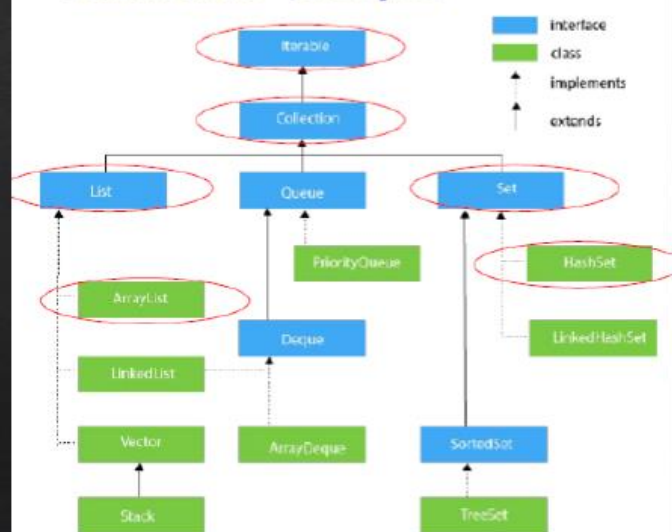
- ♦ Evita que el programa se rompa por un dato incorrecto o inexistente
- ♦ Normalmente se usa para leer datos o conexiones


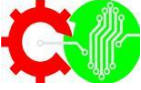
```
Path path = Paths.get( first:ruta);  
List<String> fh = new ArrayList();  
try {  
    fh = Files.readAllLines(path);  
} catch (Exception e) {  
    System.out.println( "No se pudo abrir archivo!\n"/* + e*/);  
}  
  
if(!fh.isEmpty()) {  
    return fh.toString();  
} else return null;
```

Colecciones

- ♦ Similares a los arrays[]
- ♦ Tamaño dinámico
- ♦ Almacenan diferentes tipos de datos
- ♦ Tienen métodos heredados
- ♦ Mas usados:
 - ♦ ArrayList
 - ♦ LinkedList
 - ♦ HashMap

Colecciones - Jerarquía



 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba			
2023		Tecnicatura Universitaria en Mecatrónica	
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios
Alumno:	Horacio Gonzalo Torres	Legajo:	108832
		Curso:	2M2


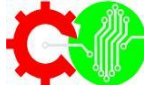
PERSISTENCIA DE LA INFORMACION

- ◆ Bases de datos
 - ◆ Bases de datos relacionales (SQL)
 - ◆ MySQL
 - ◆ PostgreSQL
 - ◆ SQLite
 - ◆ MariaDB
 - ◆ Bases no Relacionales
 - ◆ MongoDB
 - ◆ XML o JSON
- ◆ Archivos
 - ◆ Texto
 - ◆ CSV
 - ◆ Imágenes / Multimedia
 - ◆ Binario

GESTION DE DEPENDENCIAS

- ◆ Optimizan el mantenimiento y lectura de una Aplicación
- ◆ Facilidad de gestión de librerías (jar)
- ◆ Facilita el control de versiones
- ◆ Automatizacion de tareas de compilación y ejecución




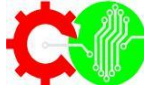
 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba		 Tecnicatura Universitaria en Mecatrónica	
2023		Pasantías y Seminarios	
Docentes	Carlos Centeno – Karina Corvalan	Materia:	
Alumno:	Horacio Gonzalo Torres	Legajo:	108832
		Curso:	2M2

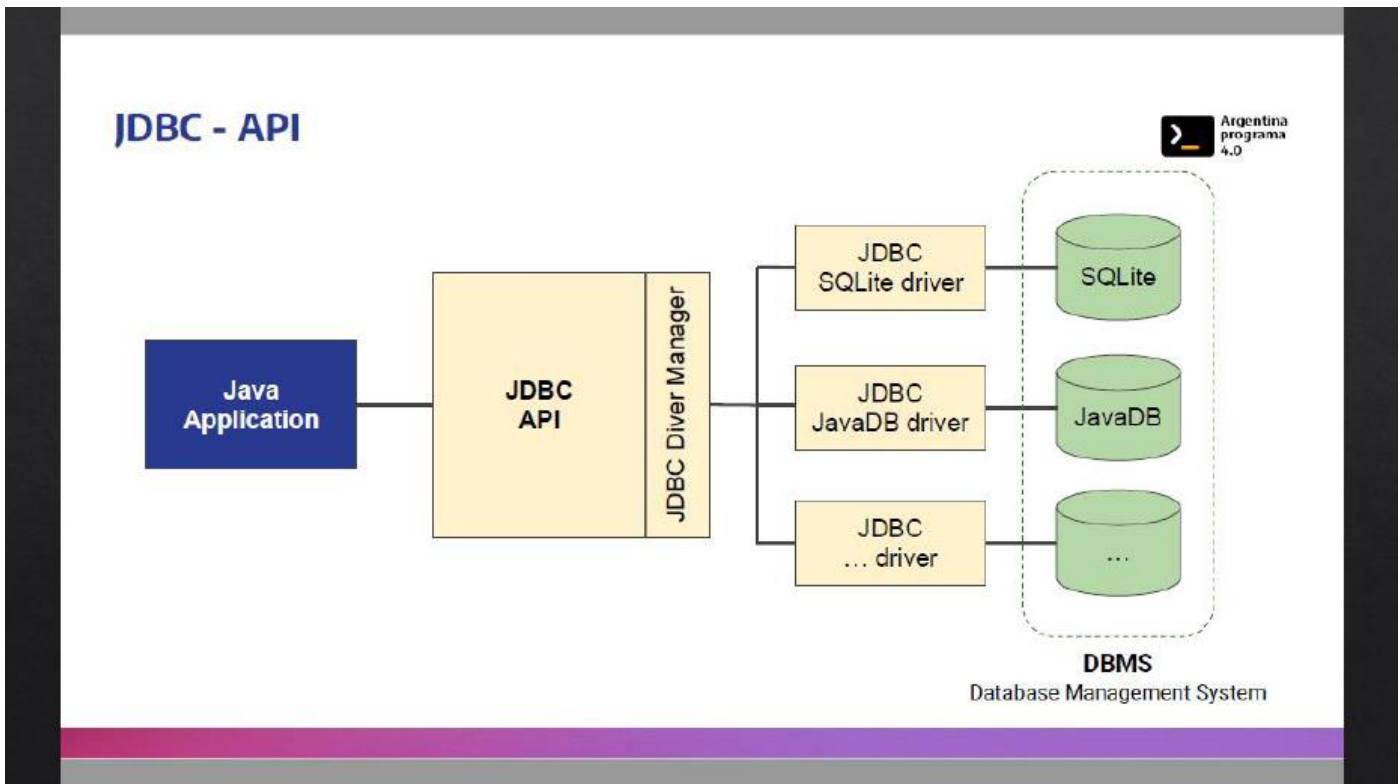
```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.utn.javaInicial</groupId>
  <artifactId>practico8</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>17</maven.compiler.source>
    <maven.compiler.target>17</maven.compiler.target>
    <exec.mainClass>com.utn.javainicial.practico8.Practico8</exec.mainClass>
  </properties>
  <dependencies>
    <!-- https://mvnrepository.com/artifact/junit/junit -->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.13.2</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

JDBC

- ♦ Java DataBase Connectivity es una API (Application Programming Interface)
- ♦ Conecta la aplicación con una base de datos
- ♦ Utiliza un Driver para conectar a la base
- ♦ Permite CRUD mediante statements
- ♦ Implementación:
 - ♦ Registrar Driver - Crear Conexión - Crear Sentencia - Ejecutar Query - Cerrar Conexion


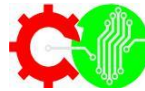
```
try {
  String jsonText = om.writeValueAsString( value: alumno.getListaMaterias());
  String sql = "INSERT INTO alumnos(legajo, nombre, materias_aprobadas) VALUES (?, ?, '"
    + jsonText + "')";
  PreparedStatement st =con.prepareStatement(sql);
  st.setInt( parameterIndex: 1, x: legajo);
  st.setString( parameterIndex: 2, x: nombre);
  st.execute();
  st.close();
  con.close();
} catch (Exception e) {
  System.out.println("Error! " + e);
}
```

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba		 Tecnicatura Universitaria en Mecatrónica	
2023		Pasantías y Seminarios	
Docentes	Carlos Centeno – Karina Corvalan	Materia:	
Alumno:	Horacio Gonzalo Torres	Legajo:	108832
		Curso:	2M2

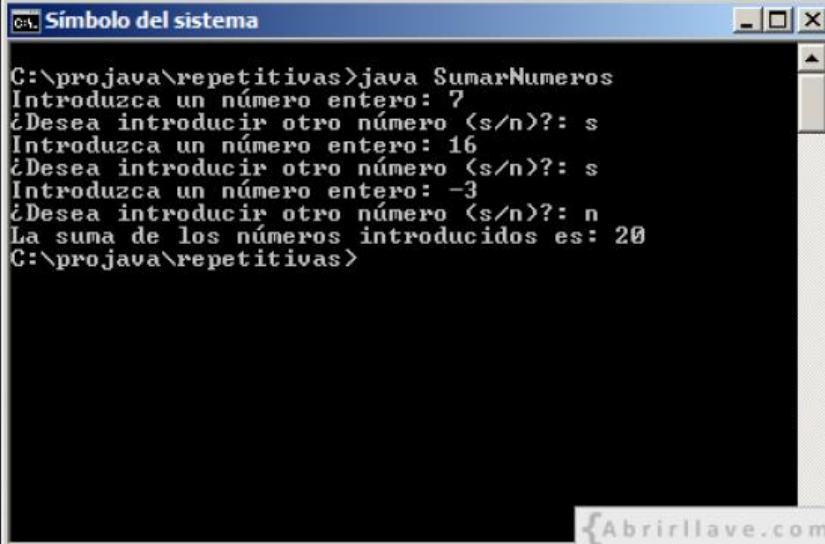


APLICACIONES

- ◆ **Consola**
 - ◆ Aplicaciones de solo texto (sin entorno gráfico)
- ◆ **Desktop Apps**
 - ◆ Winforms
 - ◆ Aplicaciones Instalables
- ◆ **Web Apps (web2)**
 - ◆ APIs
 - ◆ Aplicaciones ejecutables desde el navegador

 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba					
2023		Tecnicatura Universitaria en Mecatrónica			
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios		
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

CONSOLA



```

C:\projava\repetitivas>java SumarNumeros
Introduzca un número entero: 7
¿Desea introducir otro número (s/n)? : s
Introduzca un número entero: 16
¿Desea introducir otro número (s/n)? : s
Introduzca un número entero: -3
¿Desea introducir otro número (s/n)? : n
La suma de los números introducidos es: 20
C:\projava\repetitivas>

```

DESKTOP



Sistema 1.0

Archivo Ayuda

Formulario


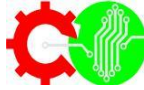
Nombre: Pedro

Apellido: Martinez

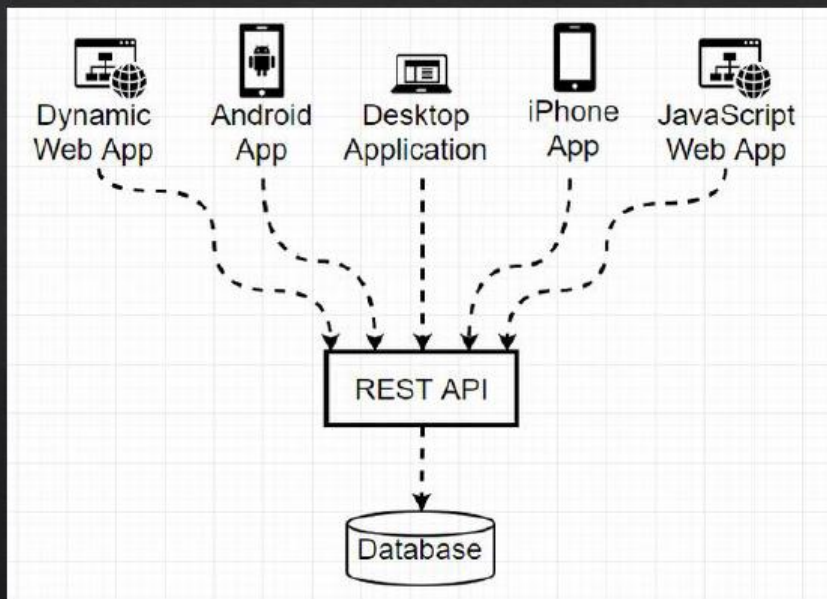
Pais Natal: Chile

Guardar Cancelar Limpiar Datos

Escuela de Informática 2013 - UNPA UACO

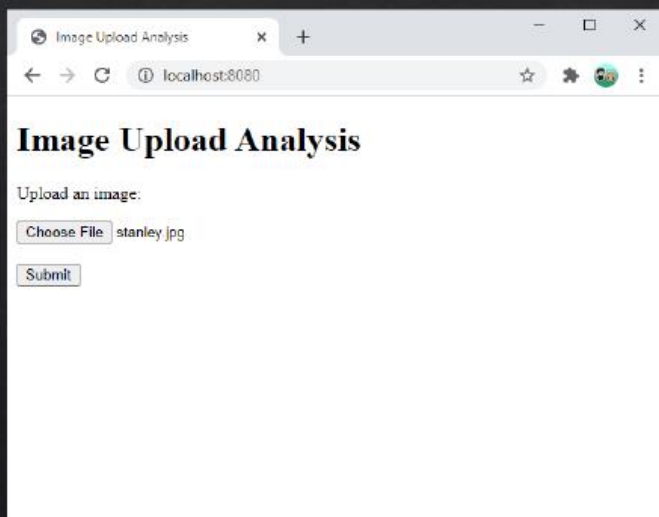
 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba		 Tecnicatura Universitaria en Mecatrónica		
2023		Pasantías y Seminarios		
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios	
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso: 2M2

API

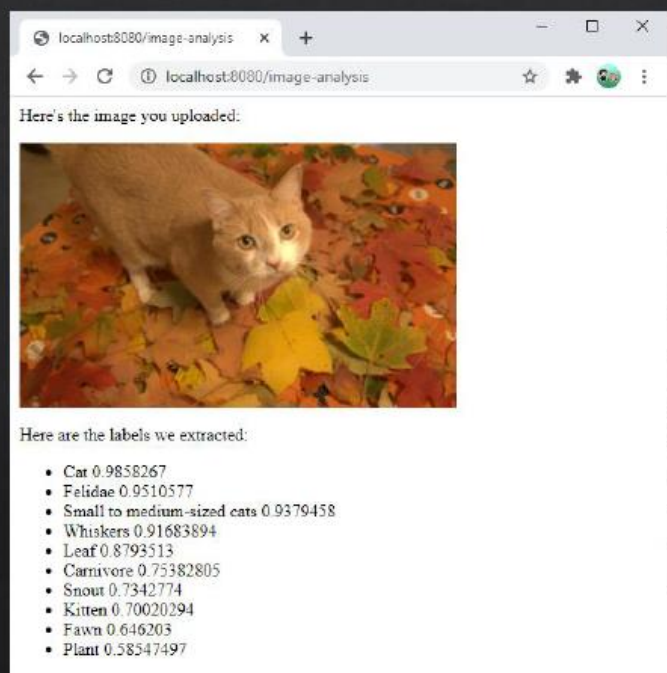



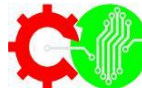
<https://happycoding.io/>

WEB APP



<https://happycoding.io/>



 UNIVERSIDAD TECNOLÓGICA NACIONAL Facultad Regional Córdoba					
2023		Tecnicatura Universitaria en Mecatrónica			
Docentes	Carlos Centeno – Karina Corvalan	Materia:	Pasantías y Seminarios		
Alumno:	Horacio Gonzalo Torres	Legajo:	108832	Curso:	2M2

RECURSOS

- ♦ Argentina Programa
- ♦ Codo a Codo
- ♦ FreeCodeCamp
- ♦ Tutoriales Web
- ♦ Alura ONE
- ♦ Tecnicaturas
 - ♦ UTN
 - ♦ ISPC
- ♦ ¡Mucha Practica!



!!!GRACIAS!!!

<https://github.com/zalotores/SeminarioJava.git>

zalotores@gmail.com

